



# CLASE 4

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

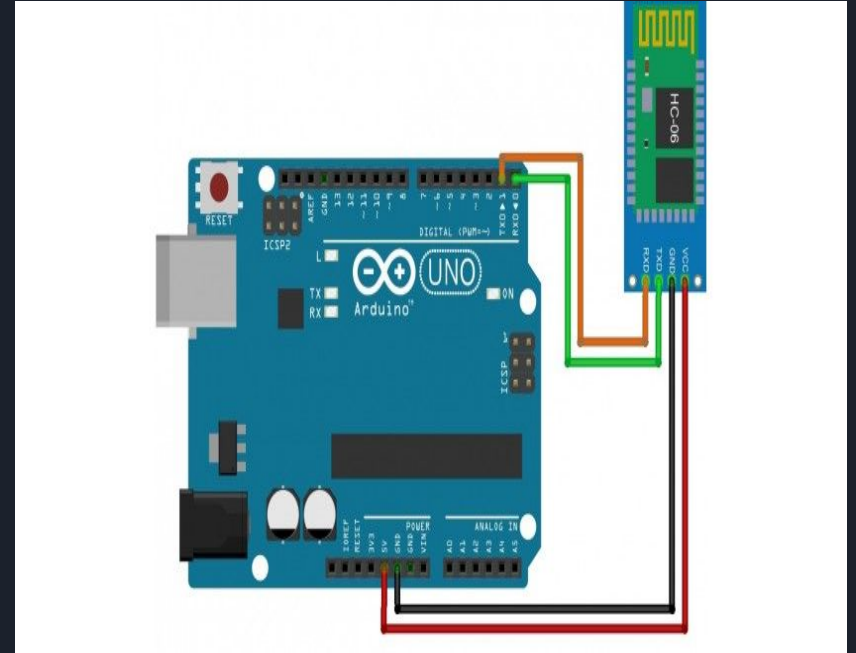


# MÓDULO BLUETOOTH

# MÓDULO BLUETOOTH

Es necesario saber que existen diferentes modelos de módulos Bluetooth entre los más populares se encuentran HC-06 y el HC-05, y entre ellos diferentes modelos.

Un módulo Bluetooth HC-06 se comporta como esclavo, esperando peticiones de conexión, Si algún dispositivo se conecta, el HC-06 transmite a éste todos los datos que recibe del Arduino y viceversa.

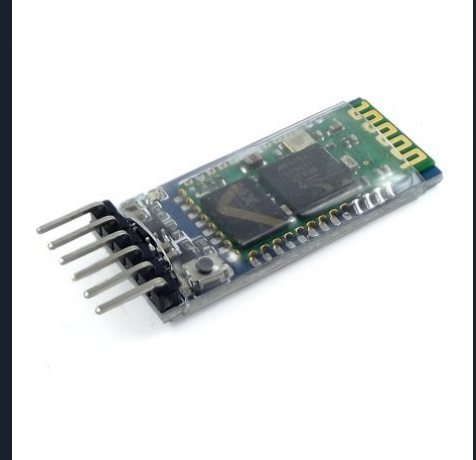
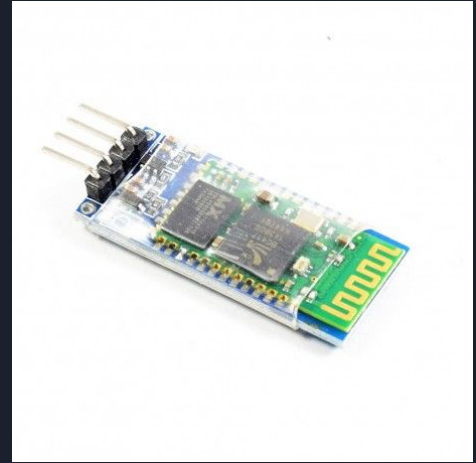


# MÓDULO BLUETOOTH

El módulo HC-06 es prácticamente idéntico a simple vista con los demás módulos que existen en el mercado.

Una simple diferencia es que el módulo HC-06 funciona como Slave y el HC-05 como Master y Slave (lo que podría confundir a algunos).

Físicamente se diferencian por el número de pines. En el HC-06 tiene un conector de 4 pines mientras que el HC-05 trae uno de 6 pines





# REGISTROS EN ARDUINO



# REGISTROS DE ARDUINO

Donde X hace referencia al número de puerto.

## REGISTROS PORTX

Es el que controla si el PIN está en un nivel alto o bajo.

## REGISTROS PINX

Este nos sirve para leer el estado de un PIN que es un input. Únicamente es de lectura no se puede escribir un valor.

## REGISTROS DDRX

Determina o establece si el PIN es una entrada o es una salida, se puede usar para leer o escribir cómo actuará el PIN si de entrada o salida.



# Ejemplo: Arduino UNO

Configurar los pines digitales del 2 al 7 como Salida.

```
void setup(){
    DDRD= B11111100;//Utilizamos D porque es del 0 al 7 y B porque es
//En este ejemplo no utilicé los pines 0 y 1 del Arduino.
/*Esta instrucción es lo mismo que hacer esto:
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(7,OUTPUT); */
}
void loop(){
```

# PIN MAPPING (Arduino UNO)

Atmega168 Pin Mapping			
Arduino function			Arduino function
reset	(PCINT14/RESET) PC6	1	28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25 PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24 PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23 PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22 GND GND
GND	GND	8	21 AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20 AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17 PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)

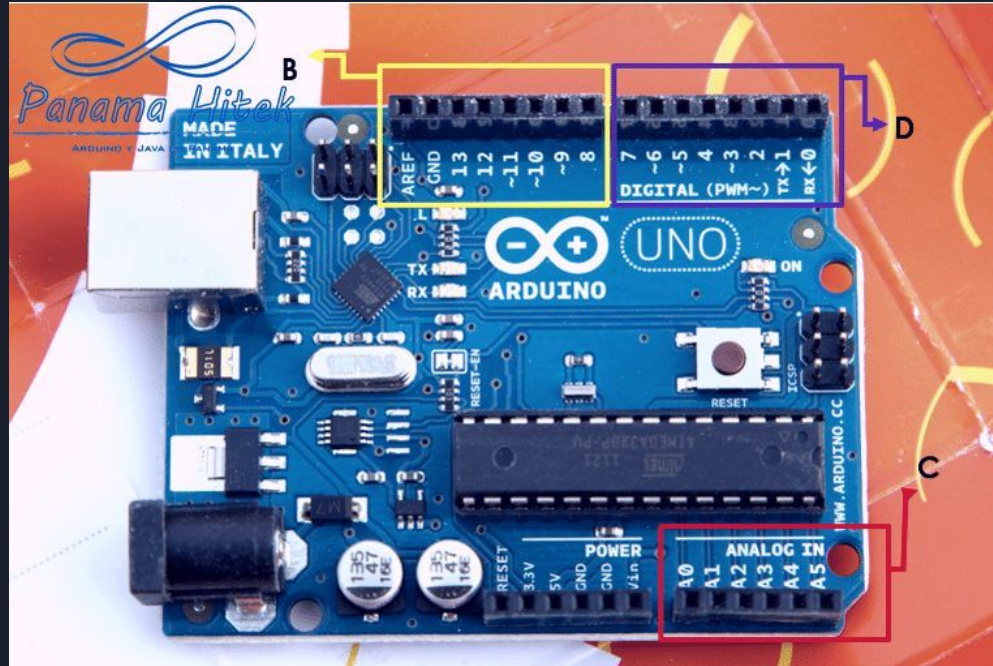
  
**Panama Hitek**

ARDUINO Y JAVA EN PANAMÁ

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.



# MAPEO DE PUERTOS (Arduino UNO)





# Ventajas

Puede que puedas cambiar los pines de estado muy rápido, en fracciones de microsegundos. Las funciones `digitalRead()` y `digitalWrite()` son cada una cerca de una docena de líneas de código, lo cual al ser compilado se convierte en unas cuantas instrucciones máquina. Cada instrucción máquina necesita un ciclo de reloj a 16 MHz, lo cual puede sumar mucho tiempo en aplicaciones muy dependientes del tiempo. El Registro PORT (Puerto) puede hacer el mismo trabajo en muchos menos ciclos de trabajo.

Algunas veces necesitamos configurar muchos pines exactamente al mismo tiempo. Por lo que usar las funciones `digitalWrite (10,HIGH)`, seguida de la función `digitalWrite (11,HIGH)`, causará que el pin 10 se ponga en nivel alto varios microsegundos después que el pin 11, lo cual puede confundir circuitos digitales conectados al Arduino, cuyo funcionamiento dependa del tiempo preciso del cambio de esos bits.



# HOJA DE TRABAJO

Realizar en parejas o individual (Utilizando únicamente registros PORT, DDR y PIN). Restringido el uso de sentencias digitalRead, digitalWrite, pinMode, analogRead, analogWrite.

Conectar dos display de 7 segmentos al arduino, directamente (sin uso de drivers). Mostrar los números de carnet (uno en cada display). Conectar un potenciómetro, al girarlo a la derecha la velocidad a la que se muestra el carnet del primer display aumenta y la del segundo display disminuye, al girarlo a la izquierda la velocidad a la que se muestra el carnet en el segundo display aumenta y la del primero disminuye.



# Referencias

## Módulo Bluetooth

[https://www.naylampmechatronics.com/blog/12\\_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html](https://www.naylampmechatronics.com/blog/12_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html)

## PIN Mapping Arduino Mega

<https://www.arduino.cc/en/Hacking/PinMapping2560>